

**REMARKS**

Reconsideration of the application in its amended form is respectfully requested.

Turning first to the rejection under 35 USC 112, the Examiner has rejected the specification under 35 U.S.C. 112, first paragraph, alleging that it fails to adequately teach the claimed limitation "C++ based application" as recited in claims 1-34. Applicants respectfully request reconsideration of this rejection for the reasons set out below.

The specification as originally filed describes on pages 12-15 "examples of application recovery systems for recovering Microsoft Foundation Classes (MFC) applications for runtime problems" (paragraph [0045]). MFC applications are C++ based applications. MFC is part of Visual C++ Libraries. Visual C++ is Microsoft's version of C++.

Furthermore, on page 14, paragraph 0058, the specification as filed discloses that C-style exceptions need to be translated into C++ style MFC exceptions "before the application can handle them". This indicates that the application is capable of handling C++ style MFC exceptions.

Accordingly, while the specification as originally filed did not explicitly use the term "C++ based application", those skilled in the art would clearly understand, upon reading of these descriptions, that the specification as filed adequately teaches the limitation of "C++ based application" capable of handling C++ exceptions, as recited in claims 1-34.<sup>1</sup>

The Examiner has also rejected claims 1-34 under 35 U.S.C. 112, first paragraph, and stated that there does not appear to be a written description of the claimed limitation of "C++ based application" in the application as filed.

---

<sup>1</sup> Paragraph [0045] has been amended to add specific reference to "C++ based applications". No new matter has been entered since it is well known to the art that MFC applications are C++ based applications. See the attached Exhibit A.

As discussed above, the limitation of "C++ based application" is described in the specification as filed in such a way as to reasonably convey to one skilled in the relevant art that the inventors, at the time the application was filed, had possession of the claimed invention.

Therefore, Applicants submit that the specification and claims comply with the requirements under 35 U.S.C. 112, first paragraph. Withdrawal of these rejections are respectfully requested.

### **35 U.S.C. 103 Rejection**

The Examiner has rejected claims 1, 3, 4, 7, 13, 14, 21, 23, 24, 26, 33 and 34 under 35 U.S.C. 103(a), stating that these claims are unpatentable over Kannan et al (US Patent No. 5,818,702, hereinafter called "Kannan") in view of Bak et al (US Patent No. 6,415,381 B1, hereinafter called "Bak"). Applicants respectfully request reconsideration of the rejection in view of the above discussion regarding the limitation of "C++ based application" and for the reasons set out below.

### **Claim 1**

As discussed in the previous response, Kannan uses a safe message loop 109 which is "a portion of code that replaces, or substitutes for this main event or message loop of any application 105 once the application generates a fatal exception" (column 4, lines 58-61). The safe message loop 109 is loaded into the memory 103 in advance (column 6, lines 36-37).

As the Examiner has acknowledged, Kannan does not teach translating the trapped exception into an exception that the application is capable of handling, nor the application is a C++ based application and the exception is a C++ exception. The Examiner has cited Bak as supplying these missing teachings.

HAYES SOLOWAY P.C.  
30 W. CUSHING STREET  
TUCSON, AZ 85701  
TEL. 520.882.7623  
FAX. 520.882.7643

175 CANAL STREET  
MANCHESTER, NH 03101  
TEL. 603.668.1400  
FAX. 603.668.8567

Bak handles exceptions in an execution stack that stores frames for functions written in multiple programming languages (column 2, lines 50-52), as discussed in the previous response.

As the Examiner has pointed out, Bak in column 13, lines 8-10 states that “a C++ exception may be thrown at a step 957 by throwing the exception stored in thread local storage. Accordingly, a Java exception has been transformed and rethrown as a C++ exception”. This statement is made in the section that describes “the process for the handler for entry frame” (column 12, lines 64-65). The entry frame “stores pointers to a previous frame on the execution stack for a function written in the same programming language” (column 2, lines 55-57). The rethrowing of a C++ exception is performed when the unshield 717 in FIG. 12 determines that an exception is pending (column 13, lines 1-8). This is an embodiment “where the Java virtual machine is written in the C++ programming language. Accordingly, the exception stack will include C++ frames and Java frames” (column 11, lines 33-36), and FIG. 12 shows the “execution stack including both C++ and Java frames” (column 11, lines 42-43). This transformation of the Java exception as a C++ exception is performed to propagate through the functions on the execution stack (column 11, lines 25-32).

This is different from translation of a trapped exception as required by claim 1 of the present application. Claim 1 recites trapping of the exception caused due to a runtime fault in a thread executing a C++ based application, and translating of the trapped exception into a C++ exception that the C++ based application is capable of handling, thus the execution of the application can be continued. In contrast, Bak does not disclose such trapping of an exception or translation of a trapped exception.

As Bak is directed to exception handling in an execution stack, there is no motivation to combine this reference with Kannan which does not disclose any execution stack.

Accordingly, Applicants respectfully submit that the teaching of Bak cannot be combined with the teaching of Kannan. Furthermore, even if one skilled in the art were to attempt to combine Bak with Kannan, he would still replace an existing message loop of an application with a pre-stored safe message loop according to the teaching of Kannan, and require an execution stack as per the teaching of Bak to throw a C++ exception when an exception is pending. However, as there is no execution stack in Kannan, he would not be able to throw a C++ exception or transform a Java exception as a C++ exception as taught by Bak. Thus, he would still fail to reach the invention as required by claim 1 of the present application.

Therefore, Applicants respectfully submit that the invention as recited in amended claim 1 has been patentably distinguished over Kannan and Bak.

### **Claims 3, 4 and 7**

Claims 3, 4 and 7 depend on claim 1 and are patentable over Kannan and Bak for the same reasons above adduced relative to claim 1, as well as for their own additional limitations (35 USC 112).

### **Claim 13**

The Examiner has rejected claim 13 referring to the rejection of claim 1.

In view that the limitation of "C++ based application" is supported by the application as originally filed, and the above discussion presented for claim 1, it is respectfully submitted that claim 13 also is patentably distinguished over Kannan and Bak, for the same reasons discussed above.

### **Claim 14**

HAYES SOLOWAY P.C.  
30 W. CUSHING STREET  
TUCSON, AZ 85701  
TEL. 520.882.7623  
FAX. 520.882.7643

175 CANAL STREET  
MANCHESTER, NH 03101  
TEL. 603.668.1400  
FAX. 603.668.8567

Claim 14 depends on claim 13. Accordingly, as discussed above, claim 14 is also patentable over Kannan and Bak for the same reasons above adduced relative to claim 1, as well as for its own additional limitations.

#### **Claim 21**

The Examiner has rejected claim 21 referring to claim 1.

Similarly to claim 13, in view that the limitation of "C++ based application" is supported by the application as originally filed, and the above discussion of claim 1, it is respectfully submitted that claim 21 has been also patentably distinguished over Kannan and Bak, for the same reasons discussed above.

#### **Claim 23**

Claim 23 depends on claim 21. Accordingly, claim 23 is also patentable over Kannan and Bak for the same reasons above adduced relative to claim 21 as well as for its own additional limitations.

#### **Claim 24**

The Examiner has indicated that it would have been obvious to combine the teaching of Kannan and Bak to improve the performance of the system of Kannan by being able to handle the exception in multiple programming languages.

In view that the limitation of "C++ based application" is supported by the application as originally filed, as discussed above in connection with claim 1, neither Kannan nor Bak discloses or suggests an exception translator as recited in claim 24. Bak's transformation of exceptions is carried out to propagate exceptions over an execution stack. Thus, it is not possible to combine Bak's transformation of exceptions into Kannan's system because there is no execution stack disclosed in Kannan. Even if one attempts to combine them, he would not

improve the performance of Kannan's system, as he would still replace the loops as taught by Kannan to handle fatal exceptions, and still require an exception stack. As Kannan does not disclose an execution stack, however, Bak's teaching cannot be incorporated in Kannan's system.

Accordingly, it is respectfully submitted that claim 24 also is patentably distinguished over Kannan and Bak.

#### **Claim 26**

Claim 26 depends on claim 24. Accordingly, claim 26 is also patentable over Kannan and Bak for the same reasons above adduced relative to claim 24, as well as for its own additional limitations.

#### **Claims 33 and 34**

These claims are similar to claim 1. In view that the limitation of "C++ based application" is supported by the application as originally filed, and the above discussion presented for claim 1, it is respectfully submitted that claims 33 and 34 have been also patentably distinguished over Kannan and Bak.

Thus, all of claims 1, 3, 4, 7, 13, 14, 21, 23, 24, 26, 33 and 34 are patentable over Kannan and Bak.

#### **35 U.S.C. 103(a) Rejection of Claims 2, 5, 6, 8, 15, 16, 22, 25, 27 and 28**

The Examiner has rejected claims 2, 5, 6, 8, 15, 16, 22, 25, 27 and 28 under 35 U.S.C. 103(a), stating that these claims are unpatentable over Kannan in view of Bak further in view of Anschuetz et al (US Patent 5,305,455, hereinafter called "Anschuetz"). Applicants respectfully request reconsideration of this rejection for the reasons set out below.

HAYES SOLOWAY P.C.  
30 W. CUSHING STREET  
TUCSON, AZ 85701  
TEL. 520.882.7623  
FAX. 520.882.7643

175 CANAL STREET  
MANCHESTER, NH 03101  
TEL. 603.668.1400  
FAX. 603.668.8567

These claims directly or indirectly depend on claims 1, 13, 21 and 24, as the case may be. The deficiencies of the combination of Kannan and Bak vis-à-vis claims 1, 13, 21 and 24 are discussed above. Anschuetz does not supply the missing teachings to Kannan and Bak to achieve or render obvious any of claims 1, 13, 21 or 24. Accordingly, the rejection of claims 2, 5, 6, 8, 15, 16, 22, 25, 27 and 28 as obvious from Kannan and Bak and Anschuetz also is in error.

**35 U.S.C. 103(a) Rejection of Claims 9-12, 17-20 and 29**

The Examiner has rejected claims 9-12, 17-20 and 29 under 35 U.S.C. 103(a), stating that these claims are unpatentable over Kannan in view of Bak further in view of Anschuetz further in view of LeVine et al (US Patent 6,591,379 B1, hereinafter called "LeVine"). Applicants respectfully request reconsideration of this rejection for the reasons set out below.

These claims also directly or indirectly depend on claims 1, 13 and 24 as the case may be. The deficiencies of a combination of Kannan, Bak and Anschuetz vis-à-vis claims 1, 13 and 24 are discussed above. LeVine does not supply the missing teachings to claims 1, 13 and 24 to achieve or render obvious Applicants' claimed invention. Accordingly, it is submitted that claims 9-12, 17-20 and 29 also are patentably distinguished over Kannan and Bak and Anschuetz further in view of LeVine.

**35 U.S.C. 103(a) Rejection of Claims 30-32**

The Examiner has rejected claims 30-32 under 35 U.S.C. 103(a), stating that these claims are unpatentable over Kannan in view of Bak further in view of Anschuetz further in view of LeVine further in view of Lillevold (US Patent 6,230,284 B1, hereinafter called "Lillevold"). Applicants respectfully request reconsideration of this rejection for the reasons set out below.

HAYES SOLOWAY P.C.  
30 W. CUSHING STREET  
TUCSON, AZ 85701  
TEL. 520.882.7623  
FAX. 520.882.7643

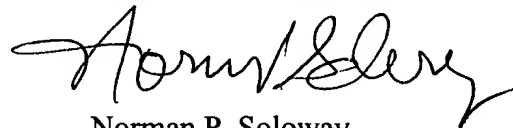
175 CANAL STREET  
MANCHESTER, NH 03101  
TEL. 603.668.1400  
FAX. 603.668.8567

These claims indirectly depend on claim 24. The deficiencies of the combination of Kannan, Bak, Anschuetz and LeVine vis-à-vis claim 24 are discussed above. Lillevold does not supply the missing teachings to achieve or render obvious claim 24. Accordingly, it is respectfully submitted that claims 30-32 have been also patentably distinguished over Kannan and Bak and Anschuetz further in view of LeVine and Lillevold for the same reasons above adduced relative to claim 24, as well as for their own additional limitations.

Having dealt with all the objections raised by the Examiner, the Application is believed to be in order for allowance. Early and favorable action are respectfully requested.

In the event there are any fee deficiencies or additional fees are payable, please charge them (or credit any overpayment) to our Deposit Account Number 08-1391.

Respectfully submitted,



Norman P. Soloway  
Attorney for Applicant  
Reg. No. 24,315

**CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: MAIL STOP AMENDMENT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on June 23, 2005, at Tucson, Arizona.

By Kim Good

NPS:kg

HAYES SOLOWAY P.C.  
30 W. CUSHING STREET  
TUCSON, AZ 85701  
TEL. 520.882.7623  
FAX. 520.882.7643

175 CANAL STREET  
MANCHESTER, NH 03101  
TEL. 603.668.1400  
FAX. 603.668.8567